

# Unidad 3

Arquitectura de Programas con Interfaz Gráfica

# Diseño de Interfaces

# Las Cuatro Actividades Básicas del Diseño

- Existen cuatro actividades básicas involucradas durante lo que es el diseño pensado en la interacción
  - Identificar las necesidades y Establecer Requerimientos
  - Desarrollar diseños que satisfagan esas necesidades
  - Construir versiones interactivas que puedan ser probadas
  - Evaluar estos prototipos para conocer que tan aceptados son

# Identificar Necesidades y Requerimientos

- Para diseñar algo que ayude a las personas, es necesario conocer hacia quien va dirigido y que tipo de apoyo o ayuda se les puede proporcionar
- El hecho de comprender las necesidades y requerimientos de los usuarios es fundamental en el inicio del proceso de diseño

# Desarrollando Diseños

- Es la parte fundamental del proceso y puede dividirse en dos sub actividades: diseño conceptual y diseño físico
- El diseño conceptual se refiere a qué debe realizar un producto y cómo debe realizarlo
- El diseño físico se refiere a los detalles del producto, por ejemplo, color, sonidos, menú, diseño de íconos, etc

# Características Fundamentales en el Proceso

- Existen tres características que se consideran fundamentales en el proceso de diseño para la interacción:
  - Atención al usuario
  - Especificar criterios de Usabilidad
  - Iteración

# Atención al Usuario

- Aunque en ocasiones se considera que ningún proceso de desarrollo puede involucrar a los usuarios, si debería ser posible que los considere en varios aspectos, lo que proporciona oportunidades para la evaluación y retroalimentación del usuario

# Especificar Criterios de Usabilidad

- Se deben especificar y documentar los criterios de usabilidad y los objetivos de experiencia del usuario desde el inicio del desarrollo del proyecto. Esto ayudará a los diseñadores a elegir entre varias alternativas de diseño y revisar el progreso conforme se avanza en el proyecto

# Iteración

- La iteración permite a los diseños realizados irse refinando a lo largo del tiempo a través de una retroalimentación. Este proceso de iteración requiere claridad por parte de los diseñadores, desarrolladores y los usuarios

# Necesidades y Requerimientos

# Identificando Necesidades y Requerimientos

- Un proyecto puede llegar a remplazar o actualizar a uno existente, o también puede ser un desarrollo completamente nuevo
- Cualquiera que sea el caso, es necesario conocer las necesidades y expectativas del usuario

# ¿Qué se Busca Conocer?

- Se pueden identificar dos objetivos:
- Conocer lo más posible sobre el usuario, su trabajo y el contexto del mismo, de tal manera que el sistema a desarrollar pueda auxiliarlo en lograr sus objetivos
- Producir, a partir de las necesidades identificadas, un conjunto de requerimientos que sirvan como una base sólida para comenzar a pensar en el diseño

# Definición de Requerimientos

- Un requerimiento es una sentencia acerca de un determinado producto que especifica que debe realizar o cómo debe funcionar

# Tipos de Requerimientos

- Se han identificado dos tipos de requerimientos, los funcionales y los no funcionales
- Los requerimientos funcionales dicen que es lo que el sistema debe realizar
- Los requerimientos no funcionales dicen que tipo de restricciones existen en el sistema y su desarrollo

# La Importancia de los Requerimientos

- La definición de requerimientos es una de las etapas de un proyecto que más problemas dan, especialmente debido a los requerimientos y objetivos poco claros
- El mal manejo de este paso, lleva a que tanto los desarrolladores como los clientes tengan problemas como frustración, pérdida de confianza en el cliente y el producto entre otras

# Establecimiento de Requerimientos

- La actividad de entender lo que un producto debe realizar ha recibido diversos nombres: recolección de requerimientos, captura de requerimientos, obtención de requerimientos, análisis de requerimientos e ingeniería de requerimientos

# Modelos de Navegación

# El Costo de la Navegación

- El abrir una ventana o una interfaz de una aplicación, siempre implica un costo cognitivo, especialmente las primeras veces
- Siempre hay que considerar las formas, los elementos y sobre todo lo que uno desea realizar en esa interfaz, esto se conoce como el cambio de contexto, incluso cuando ya se ha trabajado en ese entorno

# El Costo de la Navegación

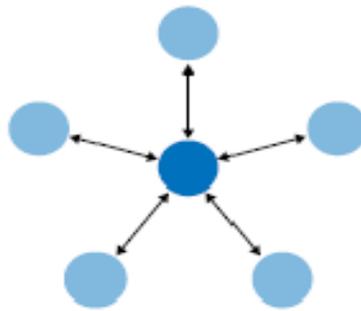
- Cada que se cambia de página o de interfaz en una aplicación hay un costo, por esta razón es necesario mantener estos costos lo más bajo posible, evitar muchos niveles de navegación llenos de diálogos y poca actividad

# Modelos de Navegación

- Un modelo de navegación se refiere a la manera en que las páginas de un sitio o en ocasiones, las interfaces de una aplicación se relacionan entre ellas

# Navegación de Estrella

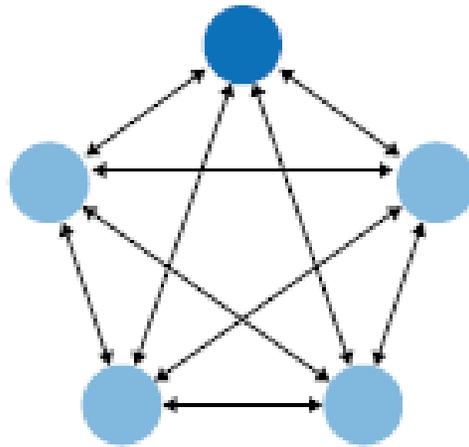
- Principalmente se encuentra en los dispositivos móviles en donde no se quieren muchos elementos en una página. Se tiene una pantalla principal que sirve como acceso a todo el resto de las pantallas o interfaces



- Para ir de una interfaz a otra, se requiere regresar a la pantalla inicial

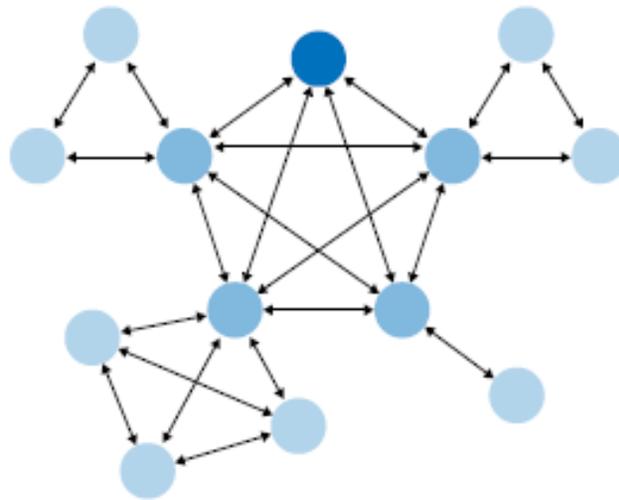
# Completamente Conectada

- Muchos sitios web siguen este modelo, hay una página o pantalla principal, pero en cada una de las otras es posible navegar y llegar a otros sitios, aunque sea solamente a un nivel



# Multinivel

- Configuración muy utilizada en los sitios web, hay páginas principales que están conectadas todas entre sí, pero las subpáginas están conectadas solamente entre ellas y a la página principal de la que descienden.



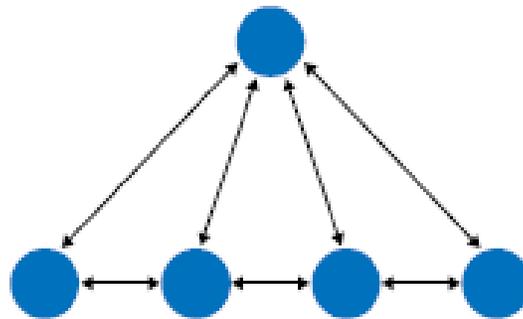
# Paso a Paso

- Configuración utilizada en las secuencias y *Wizards* en donde solo se puede avanzar hacia un estado siguiente o regresar a un estado anterior



# Navegación Piramidal

- Una variante de la navegación Paso a Paso, aquí se tiene una interfaz principal a la que se puede regresar desde cualquiera de los pasos

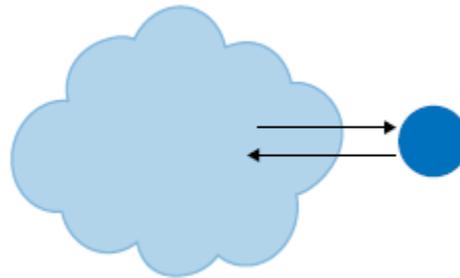


# Navegación en Tiempo y Posición

- Aplicable principalmente a elementos especiales, en donde se puede avanzar a través de ellos con un zoom o recorriendo una barra de scroll o de tiempo, dos buenos ejemplos son documentos muy largos y videos o música

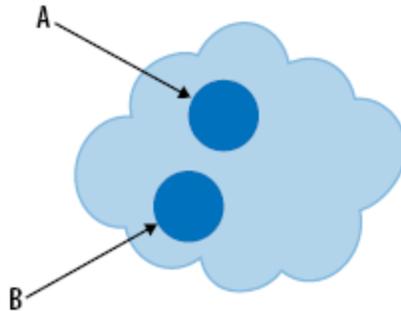
# Otros Tipos de Navegación

- Navegación Plana. En donde no hay navegación, por ejemplo un entorno de trabajo en donde se tienen disponibles todas las paletas a la vez
- Panel Modal. Son las interfaces o pantallas en donde el usuario no puede realizar actividades más allá de leer información, introducir datos o cerrarlas, un ejemplo básico es una pantalla de login



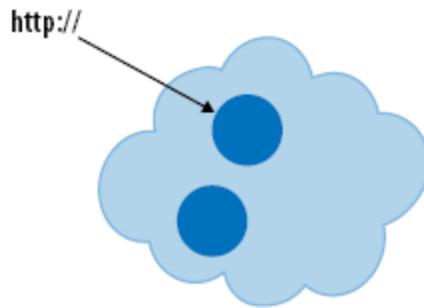
# Otros Tipos de Navegación

- Puntos de Entrada. Son interfaces que aparecen e indican al usuario la primer tarea a realizar, especialmente se utilizan para reducir el tiempo de aprendizaje. Un ejemplo básico son las entradas a ciertas aplicaciones que ofrecen las opciones básicas para el usuario.



# Otros Tipos de Navegación

- Marcadores. Se utilizan como puntos de entrada elegidos por el usuario, a los que puede llegar de manera inmediata, los marcadores en los navegadores son un claro ejemplos



# Diseño y Construcción

# Introducción

- Una vez que se tienen definidos los requerimientos, se comienza con las actividades de diseño, en general se tienen dos tipos de diseño: conceptual y físico
- El diseño conceptual se relaciona con el desarrollo de un modelo que captura lo que el producto realizará y su funcionamiento
- El diseño físico se enfoca en pantallas, imágenes, menús, íconos y gráficas

# Construcción de Prototipos

- Habiendo recolectado información acerca de lo que los usuarios desean y lo que no desean, es momento de construir prototipos e iterar sobre su desarrollo varias veces. Entre más iteraciones se tengan, mejor será el producto final

# ¿Qué es un Prototipo?

- Un prototipo no necesariamente debe ser un software, también puede ser incluso un dibujo en una hoja de papel o capturas de pantallas

# Importancia de los Prototipos

- Los prototipos son útiles cuando se discuten ideas con las personas interesadas en el desarrollo, y se considera como una de los aspectos más importantes de cada desarrollo
- Los prototipos responden a varios cuestionamientos y ayudan a los diseñadores a encontrar diferentes alternativas

# Prototipos de Baja Fidelidad

- Un prototipo de baja fidelidad es aquel que no se ve muy parecido a lo que será el producto final. Por ejemplo en cuanto al uso de materiales o diseño de interfaces
- Son útiles debido a que son simple, baratos y fáciles de producir y de modificar de tal manera que pueden ser analizados y recibirán diversos aportes de ideas y diseños alternativos

# Storyboard

- Es uno de los ejemplos de baja fidelidad, consiste principalmente en una serie de dibujos en donde se muestra el proceso que realizará un usuario al momento de interactuar con el sistema para completar una tarea

# Ventajas y Desventajas

- Ventajas
  - Menor costo de desarrollo
  - Se pueden evaluar múltiples conceptos
- Desventajas
  - Pocas oportunidades de revisar errores
  - Poca especificación de código
  - Usabilidad limitada
  - Flujo y navegación limitados

# Prototipos de Alta Fidelidad

- Los prototipos de alta fidelidad usan materiales que se esperan en el producto final y que son muy parecidos a lo que se espera como resultado

# Ventajas y Desventajas

- Ventajas
  - Funcionalidad casi completa
  - Usado para exploración y prueba
  - Herramienta para mercadeo y ventas
  - Enfocado al usuario
- Desventajas
  - Costo más elevado en desarrollo
  - Mucho tiempo de desarrollo
  - No es efectivo para recolección de requerimientos

# Uso de Prototipos de Baja Fidelidad

- Se han identificado algunas cuestiones por las que no se recomienda comenzar con prototipos de alta fidelidad
- Toman mucho tiempo en su construcción
- Renuencia a modificar aspectos que llevaron horas desarrollar
- Un error en el prototipo puede causar ansiedad
- Más complejo de evaluar y de opinar sobre él

# Tipos de Diseño de Prototipos

- Dependiendo del tipo de producto en el que se quiere trabajar.
- En prototipos de software, se puede tener una velocidad de funcionamiento menor, algunos bugs pequeños son permitidos, los íconos pueden no ser los definitivos

# Diseño Conceptual

- El diseño conceptual tiene como objetivo transformar los requerimientos del usuario y sus necesidades a un modelo conceptual el cuál se define como “una descripción del sistema propuesto en términos de un conjunto de ideas sobre lo que el sistema debe hacer y cómo debe lucir”

# Diseño Conceptual, Escenarios y Prototipos

- Un prototipo es útil para evaluar algún aspecto del proyecto que se está desarrollando
- Es conveniente tener un escenario que acompañe al prototipo que se está trabajando
- Ahí se especificará la funcionalidad esperada en uno de los pasos que componen la tarea

# Diseño Físico

- El Diseño Físico involucra tener algo más concreto, como interfaces, colores, iconos a utilizar o cómo se estructurarán los menús
- Algunas de las guías para un diseño físico son:
  - Buscar la consistencia
  - Habilitar atajos para el usuario
  - Ofrecer información de retroalimentación
  - Ofrecer mensajes de error

# Pensando en el Diseño

- Considerar los siguientes elementos:
  - Posición y estilo de los menús. En donde van a estar, que opciones tendrán y cómo se agruparán
  - Imágenes de los iconos que se pueden tener. Iconos atractivos y representativos
  - Diseño de la pantallas

# Diseño de Pantallas

- Se consideran dos aspectos en el diseño de pantallas: cómo se realiza una tarea a través de la secuencia de pantallas y la manera en que se diseña cada una de forma individual
- La primera se realiza teniendo en mente la tarea y dividiéndola en varias subtareas más pequeñas
- Una vez que se han identificado las subtareas, se analizan para ver cuál de ellas tendrá su propia pantalla
- En base a esto, se comienza con un diseño personalizado de cada una de las pantallas

# Estructura de la Interfaz

# Introducción

- El diseño de una interfaz o su *layout* tiene por objetivo manipular la atención del usuario de tal manera que se centre en algunos elementos o secuencias.

# Consideraciones Básicas de Diseño

- Existen varios elementos que se consideran básicos al momento de realizar un diseño de una interfaz:
  - Jerarquía Visual
  - Flujo Visual
  - Despliegues Dinámicos

# Jerarquía Visual

- La jerarquía visual es un concepto fundamental, aunque sencillo, significa simplemente hacer que el contenido más importante destaque sobre el resto
- Los títulos deben diferenciarse de subtítulos o del contenido en general, las listas de elementos deben parecer una lista
- Un usuario debe ser capaz de distinguir la estructura del sitio a partir de su diseño

# Implementación de Jerarquía

- Se debe considerar qué es lo más importante que se se quiere mostrar en la interfaz y lograr que la atención se centre en ese elemento
- Se deben ordenar los elementos de tal manera que la importancia vaya disminuyendo conforme van apareciendo en la interfaz
- Una buena jerarquía visual de inmediato permite aclarar: la importancia de los elementos, y las relaciones entre ellos

# Elementos de la Jerarquía

- Elementos que permiten lograr la jerarquía de distintos elementos son: densidad, color de fondo, tamaño y posición y ritmo.
- Densidad. Un bloque con una mayor densidad que el resto de los elementos que lo rodean llamará la atención sobre el resto
- Color de Fondo. El contraste entre el color de fondo y el texto llamará la atención de un elemento, el color de fondo nunca deberá opacar o dificultar al contenido que contiene

# Elementos de Jerarquía

- **Posición y Tamaño.** Un bloque de texto que resalte en la parte superior de la interfaz, de inmediato llamará la atención, mientras que uno en la parte baja y con un tamaño pequeño pasará desapercibido
- **Ritmo.** Es importante mantener un ritmo que mantenga la atención del usuario, esto incluye listas, tablas, textos e incluso espacios en blanco que permitan descansar visualmente al usuario

# Resaltando Elementos Pequeños

- Los elementos pequeños que se desean resaltar, pueden colocarse en la parte superior de la interfaz, en la parte lateral izquierda o en la esquina superior derecha
- El uso de elementos para lograr una jerarquía también puede utilizarse con elementos de pequeños, pero que pueden tener un significado o función importante

# Relación entre Elementos de la Interfaz

- Si se tienen varios elementos, a los que se les quiere dar la misma importancia, estos deberán tener el mismo aspecto y manejar un ritmo adecuado para crear una línea visualizar
- Esto se puede aplicar a párrafos, listas de elementos, menús
- Una indentación significa una relación de padre e hijo

# Programación Orientada a Eventos

# Definición

- Programación Orientada a Eventos (*Event-Driven Programming*)
- Es el modelo utilizado en el desarrollo de Interfaz Gráfica de Usuario (*Graphical User Interface*)
- La secuencia u orden en el que se ejecutan las acciones se determina por los eventos que ocurren en él
- A éstas aplicaciones se les conoce como Reactivas

# Eventos

- Un evento se define como una señal de que ha ocurrido algo y se clasifican en Externos e Internos
- Eventos Externos. Son los que se producen fuera del sistema generado por un usuario / sensor
- Eventos Internos. Son aquellos producidos por un sistema (Temporizadores, Procesos Automáticos)

# Eventos Asíncronos

- Muy relacionados con las aplicaciones gráficas
- Un evento asíncrono es aquel que ocurre fuera de la ejecución y puede suceder en cualquier momento, en este caso suceden en una interfaz de usuario

# Tipos de Eventos en GUI

- En una interfaz pueden ocurrir los siguientes tipos de eventos:
  - Componente. Cambia la estructura del componente
  - Enfoque (*Focus*). Cuando se “enfoca” un componente
  - Teclado (*Key*). Cuando se presiona una tecla

# Tipos de Eventos en GUI

- Ratón (*Mouse*). Cuando se realiza una acción con el ratón
- Ventana (*Window*). Cuando se realiza una acción sobre una ventana
- Acción (*Action*). Cuando se realiza una acción sobre un elemento

# Arquitectura Dirigida por Evento

# Definición

- Una Arquitectura Dirigida por Eventos (*EDA – Event-Driven Architecture*) es un patrón que fomenta la producción, detección y reacción a los eventos que suceden en una aplicación

# Capas de la Arquitectura

- Las capas de la EDA son:
  - Generador de Eventos
  - Canal del Evento
  - Motor de Procesamiento de Evento
  - Actividad Generada por el Evento

En la Implementación

# Etapa de Diseño

- El diseñador de éste tipo de aplicaciones debe definir los eventos que manejará la aplicación y las acciones que se realizan cuando ocurre cada uno de ellos

# Funcionamiento

- Cuando se ejecuta una aplicación desarrollada bajo este modelo, después de algunas inicializaciones el programa queda en espera de que ocurran eventos

# Propiedades

- Una propiedad es una característica de un objeto o elemento
- Las propiedades se pueden especificar al momento de la creación de la interfaz o al momento de ejecución

# Métodos

- Los Métodos son funciones que se ejecutan cuando se produce un evento sobre un elemento
- La mayoría de los lenguajes de programación ya especifican sus propios métodos y se llaman en tiempo de ejecución

# Características en la Implementación

- Contiene cuatro características fundamentales:
  - Formas y Objetos
  - Ciclos de Eventos
  - Eventos Disparadores
  - Manejadores de Eventos

# Características en la Implementación

- Formas y Objetos
  - Una Forma es un contenedor que agrupa a los controles (Objetos) que un usuario puede utilizar
- Ciclos de Eventos
  - Elementos del lenguaje de programación que identifican el evento ocurrido

# Características en la Implementación

- Eventos Disparadores
  - Utilizados por los ciclos para determinar que acción se realizará cuando ocurre un evento
- Manejador de Eventos
  - Contiene el código que se ejecutará cuando se dispare un evento

# Manejadores de Eventos

- Son pequeñas porciones de código que se ejecutan cuando ocurre un evento en particular
- Normalmente se conocen como sub rutinas ya que son parte de un programa más grande